

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Enhancement of Capability in Probabilistic Risk Analysis by Genetic Algorithms

Napat Harnpornchai

*College of Art, Media, and Technology, Chiang Mai University  
Thailand*

## 1. Introduction

Probabilistic Risk Analysis (PRA) has been widely recognized for its crucial role in various disciplines. The investigation and assessment of system failure or system malfunction is a main interest in PRA. A system is considered in a failure status when the system cannot satisfy a set of prescribed performance criteria. The prescribed performance criteria are referred to as the performance functions. The performance functions are explicitly or implicitly defined in terms of random variables that characterize the problems of interest. The performance functions also divide the high dimension space of random variables into a failure domain and a safe domain. The failure domain is the subspace of random variables that result in the failure of system. The complementary subspace is the safe domain.

The essential information that is particularly desired from PRA includes Point of Maximum Likelihood (PML) in failure domain and the failure probability. PML represents the combination of variable magnitudes that most likely contribute to the failure and to the corresponding failure probability. In practice, systems under consideration can be highly non-linear and large. It is also possible that the performance functions of systems are implicit, non-linear, non-differentiable, noisy, and can only be characterized in terms of numerical values. Computation of failure probabilities under such situations of complex systems and complicated failure domains thus generally demands considerable computational efforts and resources in order to obtain results with high confidence levels, especially in case of rare-event analysis.

The objective here is to show how Genetic Algorithms (GAs) can enhance the capability in PRA for numerous key aspects of risk-based information. Fundamental problems in PRA will be first addressed. GAs in context of PRA is next described. The demonstration of capability enhancement then follows. The demonstration starts from the application of GAs to the determination of PML. A generic problem in which it is not possible to visualize the failure domain due to its dimensionality and non-linearity characteristics is considered. The capability in PRA is significantly enhanced by GAs in such a way that the PRA can be accomplished without the requirement of a priori knowledge about the geometry of failure domain. The enhancement of the capability in PRA of rare events is subsequently illustrated. A numerical technique which utilizes the GAs-determined PML is introduced. The technique is referred to as an Importance Sampling around PML (ISPML). ISPML computes the failure probabilities of rare events with a considerably less computational effort and resource than Monte Carlo Simulation (MCS). In this regards, GAs enhance the capability in

Source: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos,  
ISBN 978-953-307-008-7, pp. 572, October 2009, I-Tech, Vienna, Austria

PRA, distinctively for rare-event analysis, through the use of considerably less computational effort and resource. Another important capability enhancement by GAs is the derivation of suboptimal Importance Sampling Functions (ISFs). The confidence levels of the computed failure probabilities become evidently improved with the utilization of suboptimal ISFs. The capability in PRA from the viewpoint of analysis accuracy is, therefore, enhanced through an aid of GAs. Afterwards, it is shown that the determination of multiple failure modes with almost equal magnitudes of likelihood in PRA can be realized with GAs. This is accomplished via the population-based search characteristics of GAs. Following the demonstration of GAs role in enhancing PRA capability, the crucial aspects of the chapter will be summarized at the end.

## 2. Fundamental problems in Probabilistic Risk Analysis (PRA)

Consider an event  $D_F$  which is defined by

$$D_F = \{\mathbf{X} | (g_1(\mathbf{X}) \leq 0) \text{ and } (g_2(\mathbf{X}) \leq 0) \text{ and } \dots \text{ and } (g_{NC}(\mathbf{X}) \leq 0)\} \quad (1)$$

More specifically, the event  $D_F$  represents a failure event of multiple and parallel minor failures.  $g_k(\mathbf{X})$  is the  $k$ -th performance function and  $NC$  is the total number of performance functions. The state of a system is defined by the performance function in such a way that

$$g_k(\mathbf{X}) = \begin{cases} \leq 0 & ; \text{fail} \\ & ; k = 1, \dots, NC \\ > 0 & ; \text{safe} \end{cases} \quad (2)$$

in which  $\mathbf{X} = [X_1 \dots X_{NRV}]^T$  is the vector of  $NRV$  random variables. Geometrically,  $D_F$  is the subspace in a multidimensional space of random variables  $X_1, \dots, X_{NRV}$  and will be referred to as the failure domain.

Each realization of  $\mathbf{X}$  in  $D_F$  in (1) represents the combination of variable magnitudes that result in the system failure. Among possible realizations of  $\mathbf{X}$ , the so-called Point of Maximum Likelihood (PML) in failure domain is of particular interest in PRA. PML represents the combination of variable magnitudes that most likely contribute to the system failure. Determination of PML is a fundamental problem in PRA.

Apart from the information about the PML, another relevant fundamental problem in PRA is the computation of the probability  $p_F$  of the failure event  $D_F$ . The failure probability  $p_F$  is crucial information and obtained from

$$p_F = \int_{D_F} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (3)$$

in which  $f_{\mathbf{X}}(\mathbf{x})$  is the Joint Probability Density Function (JPDF) of  $X_1, \dots, X_{NRV}$ .

PML yields the highest value of the JPDF in failure domain. The region in the neighborhood of PML naturally contributes to the failure probability more than the other regions in the failure domain. Such a region is referred to as an importance region. Consequently, the information about PML is highly valuable in the computation of failure probability, in addition to the characteristics of failure event. It will be shown via the numerical examples in subsequent sections that the incorporation of PML information to the computation of failure probability will considerably improve the computational efficiency. These applications include the PRA of rare events and the derivation of suboptimal ISFs.

In the next section, the application of Genetic Algorithms (GAs) to PRA will be described. The description is aimed at using GAs for solving the fundamental problems in PRA, i.e. the determination of PML and the computation of failure probability.

### 3. Genetic Algorithms (GAs) in PRA

#### 3.1 General

The fundamental and other problems in PRA can be formulated in forms of optimization problems. The optimization problems include constrained and unconstrained optimization problems. The constrained optimization problem for maximizing an objective function is expressed as

$$\text{Maximize} \quad O_1(\mathbf{x}) \quad (4)$$

$$\text{Subject to} \quad g_1(\mathbf{x}) \leq 0 \quad (5.1)$$

...

$$g_k(\mathbf{x}) \leq 0 \quad (5.k)$$

...

$$g_{NC}(\mathbf{x}) \leq 0 \quad (5.NC)$$

, where  $O_1(\mathbf{x})$  is the objective function of  $\mathbf{x} = [x_1 \quad \dots \quad x_{NRV}]^T$ .  $x_j$  is the realization of the  $j$ th random variable. The constrained maximization problem appears in the determination of PML.

The constrained minimization problem which appears in the determination of multiple design points reads

$$\text{Minimize} \quad O_2(\mathbf{x}) \quad (6)$$

$$\text{Subject to} \quad g_1(\mathbf{x}) \leq 0 \quad (7.1)$$

...

$$g_k(\mathbf{x}) \leq 0 \quad (7.k)$$

...

$$g_{NC}(\mathbf{x}) \leq 0 \quad (7.NC)$$

$O_2(\mathbf{x})$  is the objective function.

Similarly, the unconstrained optimization for minimization an objective function is expressed as

$$\text{Minimize} \quad O_3(\mathbf{x}) \quad (8)$$

, where  $O_3(\mathbf{x})$  is the objective function. Such an unconstrained minimization problem is found in the derivation of suboptimal ISFs. Each optimization problem above will be written in a more specific form for each particular PRA problem.

The objective functions of the forms represented by expressions (4), (6), and (8) are generally nonlinear. The constraints (5) and (7) represent the performance functions (2). The performance functions in practical PRA are generally implicit functions of random variables comprising a high dimensional space. In addition, the performance functions can be nonlinear, non-differentiable, noisy, and can only be characterized in terms of numerical values. There can also simultaneously be several numbers of parallel performance functions.

The operational features and solution capabilities of GAs suggest that the algorithms can effectively cope with those prescribed problem characteristics and requirements. Consequently, GAs are considered a potential tool for crucial problems in PRA. The following subsections contain the GAs elements in context of PRA application.

### 3.2 Chromosome representation

GAs work in two spaces alternatively. The selection process is performed in the space of original variables while the genetic operations are done in the space of coded variables. Both spaces are referred to as solution and coding space, respectively (Gen & Cheng, 1997). GAs encrypt each trial solution into a sequence of numbers or strings and denote the sequence as a chromosome. A simple binary coding for real values as proposed by (Michalewicz, 1996) is employed for representing chromosomes. According to the utilized coding scheme, each realization of the  $j$ th random variable  $X_j$  in the solution space is represented by a binary string as shown in Figure 1. The combination of these strings forms a chromosome in the coding space. The evaluation of chromosome fitness is done in the solution space of  $X_j$  while the genetic operations are performed in the coding space of chromosome.

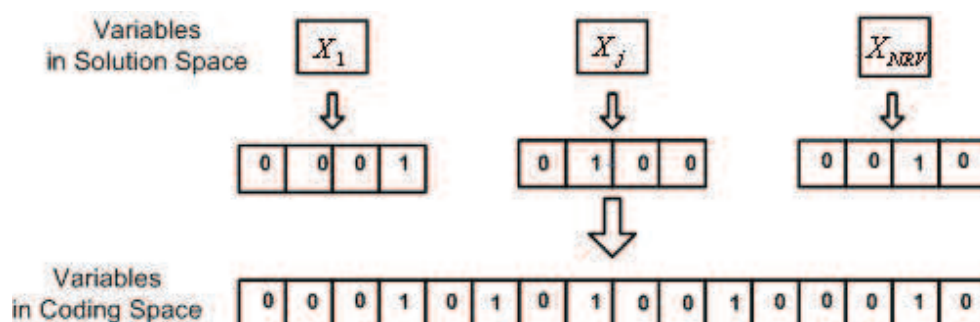


Fig. 1. Chromosome representation using binary coding for real values (Michalewicz, 1996).

### 3.3 Reproduction process

Reproduction in GAs is a process in which individual chromosomes are reproduced according to their fitness values. Fitness in an optimization by GAs is defined by a fitness function. Based on the optimization problem as described by Eq. (4) and the set of constraints (5), the fitness function  $F(x)$  of a chromosome representing a vector  $x$  of variables in the solution space is defined as

$$F(x) = \begin{cases} O_1(x) & ; x \text{ is feasible} \\ O_1(x) - \sum_{j=1}^{NC} k_j v_j(x) & ; x \text{ is infeasible} \end{cases} \quad (9)$$

The fitness function for the constrained minimization problem defined by (6) and (7) is

$$F(\mathbf{x}) = \begin{cases} 1/O_2(\mathbf{x}) & ; \mathbf{x} \text{ is feasible} \\ 1/\left[O_2(\mathbf{x}) + \sum_{j=1}^{NC} k_j v_j(\mathbf{x})\right] & ; \mathbf{x} \text{ is infeasible} \end{cases} \quad (10)$$

Note that the penalty term in this minimization case is added to the objective function. An adaptive penalty scheme which is introduced by (Barbosa & Lemonge, 2003) and improved by (Obadage & Hampornchai, 2006) will be employed to handle the constraints. The improved adaptive penalty scheme shows its excellent capability in handling a very large number of constraints (Harnpornchai et al., 2008). This adaptive scheme is given by

$$k_j = \left| \max(O_1^{\text{inf}}(\mathbf{x})) \right| \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^{NC} [\langle v_l(\mathbf{x}) \rangle]^2} \quad (11)$$

, where  $\max(O_1^{\text{inf}}(\mathbf{x}))$  is the maximum of the objective function values at the current population in the infeasible region,  $v_j(\mathbf{x})$  is the violation magnitude of the  $j$ th constraint.  $\langle v_j(\mathbf{x}) \rangle$  is the average of  $v_j(\mathbf{x})$  over the current population.  $k_j$  is the penalty parameter for the  $j$ th constraint defined at each generation. The violation magnitude is defined as

$$v_l(\mathbf{x}) = \begin{cases} |g_l(\mathbf{x})| & ; g_l(\mathbf{x}) > 0 \\ 0 & ; \text{otherwise} \end{cases} \quad (12)$$

The reproduction operator may be implemented in a number of ways. The easiest and well-known approach is the roulette-wheel selection (see e.g. (Goldberg, 1989 and Deb, 1995)). According to the roulette-wheel scheme, the  $k$ th chromosome will be reproduced with the probability of

$$P_k = \frac{F_k}{\sum_{l=1}^{NPop} F_l} \quad (13)$$

, in which  $N_{Pop}$  is the population size. The fitness value  $F_k$  is obtained from either Eq. (9) or (10). Note that subscript  $k$  in  $F_k$  signifies that the fitness value is computed for each respective  $k$ th chromosome. It is interesting to note that GAs utilize only the numerical values of the objective function and of its associated constraints for the evaluation of the chromosome fitness, as seen from Eqs. (9) – (12). This advantageous feature makes GAs readily applicable to real-world problems where the performance functions are generally implicit with respect to random variables.

### 3.4 Genetic operators

In accordance with the binary representation of chromosomes, a simple binary crossover is applied (confer Figure 2).

The mutation operation also assists the exploration for potential solutions which may be overlooked by the crossover operation. According to the chromosome representation, a binary mutation is employed for the purpose (confer Figure 3).



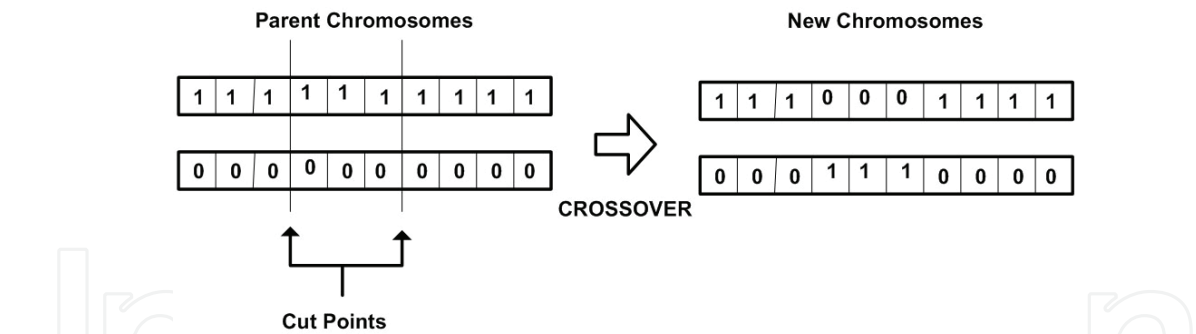


Fig. 2. Crossover of two chromosomes.

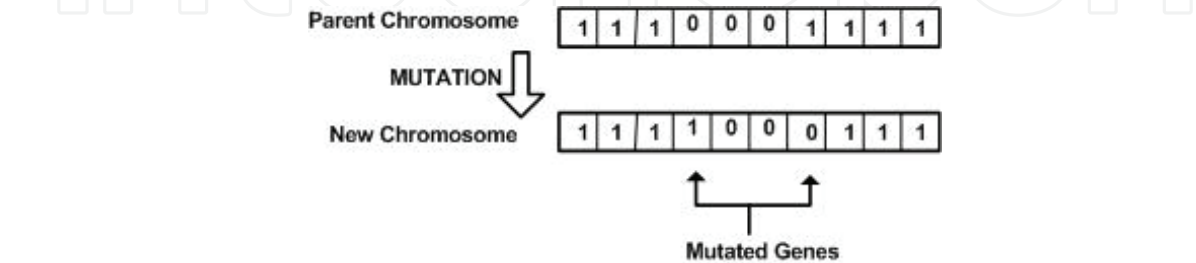


Fig. 3. Mutation on a chromosome.

3.5 Multimodal GAs

Simple GAs perform well in locating a single optimum but face difficulties when requiring multiple optima (see e.g. (De Jong, 1975; Mahfoud, 1995a; Mahfoud, 1995b and Miller & Shaw, 1995)). Niching methods can identify multiple solutions with certain extent of diversity (Miller & Shaw, 1995). Among niching methods, Deterministic Crowding Genetic Algorithms (DCGAs) (Mahfoud, 1995a and Mahfoud, 1995b) have been commonly used in multimodal functions optimization. It is noted that DCGAs is originally designed for unconstrained optimization problems. The adaptive penalty described in the previous subsection will be used in conjunction with DCGAs to handle constraints. DCGAs work as follows. First, all members of population are grouped into  $NPop/2$  pairs, where  $NPop$  is the population size. The crossover and mutation are then applied to all pairs. Each offspring competes against one of the parents that produced it. For each pair of offspring, two sets of parent-child tournaments are possible. DCGAs hold the set of tournaments that forces the most similar elements to compete. The following provides a pseudo code of DCGAS (Brownlee, 2004).

- $NPop$  : Population size.  
 $d(x, y)$  : Distance between individuals  $x$  and  $y$ .  
 $F(x)$  : Fitness of individual population member.

1. Randomly initialize population.
2. Evaluate fitness of population.
3. Loop until stop condition:
  - a. Shuffle the population.
  - b. Crossover to produce  $NPop/2$  pairs of offspring.
  - c. Apply mutation (optional).
  - d. Loop for each pair of offspring:
    - i. If  $(d(\text{parent1}, \text{child1}) + d(\text{parent2}, \text{child2})) \leq (d(\text{parent2}, \text{child1}) + d(\text{parent1}, \text{child2}))$ .

1. If  $F(\text{child1}) > F(\text{parent1})$ , child1 replaces parent1.
2. If  $F(\text{child2}) > F(\text{parent2})$ , child2 replaces parent2.
- ii. Else
  1. If  $F(\text{child1}) > F(\text{parent2})$ , child1 replaces parent2.
  2. If  $F(\text{child2}) > F(\text{parent1})$ , child2 replaces parent1.

In the following section, the applications of GAs to enhance the capability in PRA for numerous key aspects of risk-based information will be demonstrated.

## 4. Roles of GAs in enhancing PRA capability

### 4.1 Determination of PML

#### 4.1.1 Problem formulation

The likelihood of each combination of variable magnitudes in contributing a failure event is an interesting issue in PRA. The focus is particular on the so-called Point of Maximum Likelihood (PML) in failure domain. PML represents the combination of variable magnitudes that most likely contribute to the failure and to the corresponding failure probability. Since PML is the point of highest JPDF in failure domain  $D_F$ , the PML  $\mathbf{x}^*$  can be obtained from solving the following optimization problem:

$$\text{Maximize} \quad O_4(\mathbf{x}) = f_X(\mathbf{x}) \quad (14)$$

$$\text{Subject to} \quad g_1(\mathbf{x}) \leq 0 \quad (15.1)$$

...

$$g_k(\mathbf{x}) \leq 0 \quad (15.k)$$

...

$$g_{NC}(\mathbf{x}) \leq 0 \quad (15.NC)$$

in which  $f_X(\mathbf{x})$  is the JPDF of  $\mathbf{X}$ .  $\mathbf{X} = [X_1 \dots X_{NRV}]^T$ .  $X_l$  is the  $l$ th random variable.  $g_k(\mathbf{x})$  ( $k = 1, \dots, NC$ ) is the  $k$ th performance function.  $N$  and  $NC$  are the total number of basic random variables and the total number of performance functions, respectively. The fitness function is defined as

$$F(\mathbf{x}) = \begin{cases} O_4(\mathbf{x}) & ; \mathbf{x} \text{ is feasible} \\ O_4(\mathbf{x}) - \sum_{j=1}^{NC} k_j v_j(\mathbf{x}) & ; \mathbf{x} \text{ is infeasible} \end{cases} \quad (16)$$

, where  $k_j$  and  $v_j(\mathbf{x})$  are defined as in Eqs. (11) and (12), respectively.

#### 4.1.2 Numerical example 1

Consider the following performance function (Lee et al. 2006)

$$g(\mathbf{Y}) = \sigma_u - \sigma_{\text{local}}^M \quad (17)$$



where  $\mathbf{Y}$  is the vector of random variables constituting the performance function (17).  $\sigma_u$  is a random variable.  $\sigma_{\text{local}}^M$  is defined as

$$\sigma_{\text{local}}^M = \frac{m_{\text{op}}}{\left(M_{\text{ref}} / \sigma_y\right)}$$

(18)

$$M_{\text{ref}} = \frac{M_L}{1.333}$$

(19)

$$M_L = 4R_m^2 t \sigma_y \left[ \cos\left(\frac{\pi D \Theta}{8t}\right) - \frac{D}{t} \frac{f(\Theta)}{2\Theta} \right]$$

(20)

$$f(\Theta) = 0.7854\Theta^2 - 0.09817\Theta^4 + 0.0040906\Theta^6 - 0.000085\Theta^8$$

(21)

The description of each random variable is given in Table 1.

Variable	Distribution Type	Mean	COV
$D$	Normal	$4.3 \times 10^{-3}$	0.10
$L$	Normal	$100 \times 10^{-3}$	0.10
$\Theta / \pi$	Normal	0.5	0.10
$D_o$	Normal	$114.3 \times 10^{-3}$	0.02
$\Sigma_y$	Log-normal	$326 \times 10^6$	0.07
$\Sigma_u$	Log-normal	$490 \times 10^6$	0.01

Table 1. Description of random variables in Numerical Example 1 and 2.

$m_{\text{op}}$  and  $t$  are deterministic variables.  $m_{\text{op}}$  is equal to  $16 \times 10^3$  whereas  $t$  is set to be  $8.6 \times 10^{-3}$ .  $R_m$  is defined as

$$R_m = \frac{R_I + R_E}{2}$$

(22)

, where

$$R_I = \frac{1}{2}(D_o - 2t)$$

(23)

$$R_E = \frac{D_o}{2}$$

(24)

More specifically,

$$\mathbf{Y} = \left[ D \quad L \quad \frac{\Theta}{\pi} \quad D_o \quad \Sigma_y \quad \Sigma_u \right]$$

(25)

GAs have been applied to determine PML. The objective function according to Eq. (14), is

$$O_5(\mathbf{Y}) = f_D(d)f_L(l)f_{\Theta/\pi}(\theta/\pi)f_{D_o}(d_o)f_{\Sigma_y}(\sigma_y)f_{\Sigma_u}(\sigma_u)$$

(26)

The magnitude of the constraint violation, according to Eqs. (12) and (17), is

$$\nu(\mathbf{Y}) = \begin{cases} \left| \sigma_u - \sigma_{\text{local}}^M \right| & ; g(\mathbf{Y}) > 0 \\ 0 & ; \text{otherwise} \end{cases} \tag{27}$$

The corresponding fitness function is thus

$$F(\mathbf{Y}) = \begin{cases} O_5(\mathbf{Y}) & ; g(\mathbf{Y}) \leq 0 \\ O_5(\mathbf{Y}) - k\nu(\mathbf{Y}) & ; \text{otherwise} \end{cases} \tag{28}$$

GAs search employs the population size of 100. The number of generations used in the search is 200. A two-point crossover is utilized with the crossover rate of 0.8. The mutation rate is taken as 0.002. The resulting PML is shown in Table 2.

PML	Magnitude at PML
$d^*$	$5.28 \times 10^{-3}$
$l^*$	$1.00 \times 10^{-1}$
$\theta^* / \pi$	$5.47 \times 10^{-1}$
$d_o^*$	$1.07 \times 10^{-1}$
$\sigma_y^*$	$3.24 \times 10^8$
$\sigma_u^*$	$3.99 \times 10^8$

Table 2. PML in Numerical Example 1.

The performance function in this example is highly nonlinear and implicit function of random variables. The performance function is also defined in terms of a mixture of different types of random variable, not only a normal type. It should be noted that the operation of GAs in determining the PML does not require prior knowledge about the problem characteristics. GAs, therefore, can enhance the capability in PRA for the determination of PML in complicate situations.

4.2 PRA of rare events by Importance Sampling around PML (ISPML)

4.2.1 Background notion

The probability  $p_F$  of the failure event  $D_F$  as defined by Eq. (3) inevitably requires computational procedures for its accurate assessment in practical problems. It is widely recognized that Monte Carlo Simulation (MCS) is the only tool that is applicable to a wide range of problems in assessing failure probability. A major drawback of MCS is that the procedure requires large sample sizes in order to compute probabilities of very low orders in PRA of rare events when demanding high confidence levels. An efficient strategy for overcoming this undesirable situation is the utilization of the so-called importance sampling (Fishman, 1996). The notion behind the importance sampling is that the procedure performs more sampling in the importance region of failure domain. From the probabilistic characteristics of PML, the region in the neighborhood of PML can be regarded as the importance region because PML yields the highest value of the JPDP in failure domain. Consequently, the importance sampling should purposely concentrate around PML. Using the importance sampling technique, Eq. (3) is modified to

$$p_F = \int I(y) \frac{f_X(y)}{h_X(y)} h_X(y) dy \quad (29.1)$$

, or

$$p_F = E_h \left[ I(Y) \frac{f_X(Y)}{h_X(Y)} \right] \quad (29.2)$$

in which  $I(Y)$  is the indicator function and is defined as

$$I(Y) = \begin{cases} 1 & ; Y \in D_F \\ 0 & ; Y \notin D_F \end{cases} \quad (30)$$

Note that the subscript  $h$  signifies that the expectation  $E$  is taken with respect to an importance sampling JPDP or Importance Sampling Function (ISF)  $h_X(x)$ . According to MCS, the failure probability is estimated as

$$P_F \cong \frac{1}{N_{sim}} \sum_{j=1}^{N_{sim}} I(Y_j) \frac{f_X(Y_j)}{h_X(Y_j)} \quad (31)$$

in which  $Y_j$  is the  $j$ th realization sampled from the ISF  $h_X(x)$  and  $N_{sim}$  is the sample size.

The PML obtained from GAs search can enhance the efficiency of MCS. The efficiency enhancement is accomplished by employing the GAs-searched PML as the sampling center of the ISF  $h_X(x)$ . This sampling scheme is denoted as an Importance Sampling using PML (ISPML). For the purpose of procedure clarity, the original JPDP  $f_X(x)$  will be rewritten as  $f_X(x | \mu = \mu_0)$  in which  $\mu$  denotes the mean vector.  $\mu_0$  is the original mean vector. According to the ISPML, the ISF  $h_X(x)$  takes the form

$$h_X(x) = f_X(x | \mu = x^*) \quad (32)$$

, where  $x^*$  is PML. That is the ISF has the same functional form as the original JPDP. The mean vector of the ISF, however, is different from that of the original JPDP and takes the PML as the mean vector. Consequently, the estimate of the failure probability is

$$P_F \cong \frac{1}{N_{sim}} \sum_{j=1}^{N_{sim}} I(Y_j) \frac{f_X(Y_j | \mu = \mu_0)}{f_X(Y_j | \mu = x^*)} \quad (33)$$

#### 4.2.2 Numerical example 2

Based on the GAs-searched PML, the ISF according to the ISPML procedure takes the form of Eq. (32), i.e.

$$\mu = [d^* \quad l^* \quad \theta^*/\pi \quad d_0 \quad \sigma_y \quad \sigma_u]^T \quad (34)$$

The ISF as defined by Eqs. (32) and (34) is used to compute the failure probability according to the performance function (17). The results are compared with MCS. The estimate of the failure probability in each MCS and ISPML methodology is based on 10 independent runs. The sample size per each run of ISPML is 1,000 whereas that of MCS is 1,000,000. Table 3

compares the numerical results from both methodologies. Note that the Coefficient of Variation of the estimate of failure probability  $P_F$  ( $COV_{PF}$ ) is defined as

$$COV_{PF} = \frac{S.D._{PF}}{\overline{P}_F}$$

(35)

$\overline{P}_F$  is the sample mean of  $P_F$ .  $S.D._{PF}$  is the sample standard deviation of  $P_F$ . Lower magnitudes of  $COV_{PF}$  signify higher confidence levels of probability estimates.

Methodology	$\overline{P}_F$	$COV_{PF}$	$N_{sim}$
MCS	1.00x10 <sup>-6</sup>	0.94	1,000,000
ISPML	1.28x10 <sup>-6</sup>	0.34	1,000

Table 3. Comparison of numerical results from MCS and ISPML.

It is obvious from Table 3 that the sample size used by ISPML is much smaller than the sample size for MCS but ISPML results in smaller order of the  $COV_{PF}$ . More precisely, MCS employs the sample sizes 1,000 times larger than ISPML does. The accuracy of ISPML is, however, remarkably higher than that of MCS.

This numerical example testifies that the computation of event probability can demand considerable computation resource, though the variable space is not of high dimension, when demanding high confidence levels of analysis results. It also shows that GAs help realize the estimation of low probabilities in the situation where the sample sizes may be prohibitively provided due to constrained computational resources. From the viewpoint of computational efficiency and accuracy, the capability in PRA of rare events is thus substantially enhanced by GAs.

4.3 Derivation of suboptimal Importance Sampling Functions (ISFs)

4.3.1 Problem formulation

ISPML defines an ISF as given by Eq. (32). However, there can be other definitions of ISF. The ideal ISF is the sampling JPDF that results in a zero-variance estimate. However, it is not possible in general to obtain such an optimal ISF because the determination of the optimal ISF depends on the underlying probability being computed (Fishman, 1996). Consequently, it is most desirable to obtain other alternative ISFs that reduce the variance of probability estimate as much as possible. Such variance-minimizing ISFs will be defined herein as suboptimal ISFs. The variance of probability estimate is denoted as  $VAR_h[P_F]$  or simply  $VAR$ . The subscript  $h$  informs again that the variance is taken with respect to the ISF  $h_X(\mathbf{y})$ . Since the ISF  $h_X(\mathbf{y})$  is unknown and needs to be determined, the variance-minimization problem for determining suboptimal ISFs is necessarily taken with respect to another pre-defined JPDF  $q_X(\mathbf{y})$ . It can be shown that the variance-minimization problem (Harnpornchai, 2007) can be formulated as

$$\underset{h}{\text{Minimize}} \quad E_q \left[ I^2(\mathbf{Y}) \frac{f_X(\mathbf{Y})f_X(\mathbf{Y})}{h_X(\mathbf{Y})q_X(\mathbf{Y})} \right]$$

(36)

The prescribed sampling function  $q_X(\mathbf{y})$  is referred to as a pre-sampling JPDF. The ISF  $h_X(\mathbf{x})$  which is obtained from the variance-minimization problem is a suboptimal ISF.

If the ISF  $h_X(x)$  can be completely defined by a vector of parameters  $\mathbf{v} = [v_1 \dots v_{NP}]^T$ , in which  $v_j$  is the  $j$ th parameter characterizing the JPDP of ISF and  $NP$  is the total number of JPDP parameters, then the variance-minimization problem (36) is specifically written as:

$$\text{Minimize}_{\mathbf{v}} E_q \left[ I^2(\mathbf{Y}) \frac{f_X(\mathbf{Y})f_X(\mathbf{Y})}{h_X(\mathbf{Y}; \mathbf{v})q_X(\mathbf{Y})} \right] \quad (37)$$

The expectation in (36) is approximated by

$$E_q \left[ I^2(\mathbf{Y}) \frac{f_X(\mathbf{Y})f_X(\mathbf{Y})}{h_X(\mathbf{Y}; \mathbf{v})q_X(\mathbf{Y})} \right] \cong \frac{1}{Npre} \sum_{k=1}^{Npre} I^2(\mathbf{Y}_k) \frac{f_X(\mathbf{Y}_k)f_X(\mathbf{Y}_k)}{h_X(\mathbf{Y}_k; \mathbf{v})q_X(\mathbf{Y}_k)} \quad (38)$$

The samples  $\mathbf{Y}_k (k = 1, \dots, Npre)$  are generated according to the pre-sampling JPDP  $q_X(\mathbf{y})$ . The corresponding variance-minimization problem becomes:

$$\text{Minimize}_{\mathbf{v}} O_6(\mathbf{v}) \quad (39)$$

, where

$$O_6(\mathbf{v}) = \frac{1}{Npre} \sum_{k=1}^{Npre} I^2(\mathbf{Y}_k) \frac{f_X(\mathbf{Y}_k)f_X(\mathbf{Y}_k)}{h_X(\mathbf{Y}_k; \mathbf{v})q_X(\mathbf{Y}_k)} \quad (40)$$

The objective function (40) can be of highly complicate nature in practical problems, e.g., highly non-linear, non-convex, and high-dimensional. The complex nature normally arises from the collective characteristic of the JPDPs that build up the objective function (40). GAs are considered a promising tool for searching the variance-minimizing parameters  $\mathbf{v}^* = [v_1^* \dots v_{NP}^*]^T$  under the circumstance of such a complicate objective function. When using GAs for the unconstrained minimization problem (39), the fitness function is defined as

$$F(\mathbf{v}) = \frac{1}{O_6(\mathbf{v})} \quad (41)$$

The following subsection illustrates how GAs are applied for deriving a suboptimal ISF.

#### 4.3.2 Numerical example 3

Consider two independent and identical random variables of normal type  $X_1$  and  $X_2$ , whose JPDP is given by

$$f_X(x_1, x_2) = \prod_{j=1}^2 (2\pi)^{-1/2} \exp\left(-\frac{x_j^2}{2}\right) \quad (42)$$

The failure event  $D_{F3}$  is defined as

$$D_{F3} = \{(X_1, X_2) | (g_1(X_1, X_2) \leq 0) \text{ and } (g_2(X_1, X_2) \leq 0)\} \quad (43)$$

The performance functions are

$$g_1(X_1, X_2) = 5 - X_1 \quad (44.1)$$

$$g_2(X_1, X_2) = 5 - X_2$$

(44.2)

A JPDPF of two identical and independent normal PDF is used as the pre-sampling PDF  $q_X(x)$ , i.e.

$$q_X(x_1, x_2) = \prod_{j=1}^2 (2\pi\sigma_{p_j})^{-1/2} \exp\left[-\frac{(x_j - \mu_{p_j})^2}{2\sigma_{p_j}^2}\right]$$

(45)

The pre-sampling is performed around the PML in failure domain. It can be shown that the PML is (5.0, 5.0) for the performance functions (44). Correspondingly, the mean vector of the pres-sampling JPDPF  $q_X(x)$  is equal to  $[5.0 \ 5.0]^T$ , i.e.,  $\mu_{p1} = \mu_{p2} = 5.0$ . The vector of the standard deviation is, however, set to that of the original JPDPF  $f_X(x)$ , i.e.,  $\sigma_{p1} = \sigma_{p2} = 1.0$ . The pre-sampling around PML utilizes the sample size of 100, i.e.  $N_{pre} = 100$ . Each realization  $Y_k = [X_1 \ X_2]^T_k$  must be checked with the performance functions (44.1) and (44.2) in order to determine the value of  $I(Y_k)$ , following

$$I(Y_k) = \begin{cases} 1 & ; g_1(Y_k) \leq 0 \text{ and } g_2(Y_k) \leq 0 \\ 0 & ; \text{otherwise} \end{cases}$$

(46)

The ISF employs the same parametric JPDPF as the original JPDPF. Correspondingly, the ISF is

$$h_X(x_1, x_2; \nu) = \prod_{j=1}^2 (2\pi\sigma_j)^{-1/2} \exp\left[-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right]$$

(47)

in which  $\nu = [\mu_1 \ \sigma_1 \ \mu_2 \ \sigma_2]^T$  is the vector of the ISF parameters to be optimized. GAs are utilized to determine the variance-minimizing parameters. The search procedure employs  $N_{pop} = 100$ , the crossover rate of 0.8, and the mutation rate of 0.002. The variance-minimizing parameters  $\nu^* = [\mu_1^* \ \sigma_1^* \ \mu_2^* \ \sigma_1^*]^T$  from the GAs search are  $[5.1 \ 0.3 \ 5.1 \ 0.3]^T$ . The suboptimal ISF with the determined variance-minimizing parameters  $[5.1 \ 0.3 \ 5.1 \ 0.3]^T$  is utilized for estimating the failure probability. The probability estimate is the average of 10 independent runs of the optimal ISF. The resulting VAR is also computed, based on those 10 computed values of probability. Each optimal ISF run employs a sample size of 1,000. The results are summarized in Table 4.

$\bar{P}_F$	VAR	$COV_{PF}$
8.33x10 <sup>-14</sup>	1.11x10 <sup>-29</sup>	0.04

Table 4. Computation of failure probability by the GAs-derived suboptimal ISF.

Although the problem considered is of low dimension, the failure probability is at extremely low order, i.e. 10<sup>-14</sup>. The magnitude of the  $COV_{PF}$  from the GAs-derived suboptimal ISF is extremely low, i.e. 0.04. In other words, the confidence level of the probability estimate is considerably high. For such an order of estimate and a confidence level, MCS requires a sample sizes at least 10<sup>16</sup>. The order of the sample size used by the suboptimal ISF is thus 10<sup>13</sup> times less that that required by MCS for the same  $COV_{PF}$  as indicated in Table 4.



This numerical example shows that GAs facilitate the derivation of suboptimal ISFs. It should be noted that the knowledge of problem is unnecessary at all for the GAs operation. Consequently, high confidence levels in PRA of complex problems becomes possible even if there is no or little a priori knowledge of the problems.

#### 4.4 Determination of multiple design points

##### 4.4.1 Problem characteristics

Design point is the point on the limit state surface that is nearest to the origin in a standard normal space. In optimization context, the design point is the global minimum obtained from solving a constrained optimization problem. However, it is possible that there are other local minima whose distances to the origin are of similar magnitudes to the global minimum. The global minimum and local minima lead to the situation of multiple design points. When multiple design points exist, PRA based only on any single design point among multiple design points may result in an underestimation of failure probability. Determination of global optimum as well as local optima leads to multiple solutions, which is classified as a multimodal optimization problem.

The following subsections intend to demonstrate how GAs enhance the capability in PRA to cope with multiple failure events or modes. Such an application is important when several failure events are almost equally critical. It will be shown that the determination of multiple design points is readily accomplished using DCGAs. The adaptive penalty technique as described in the afore-mentioned subsection will be combined with DCGAs for handling constraints. This is a novelty because multimodal GAs were originally designed for unconstrained multimodal optimization.

From the definition of the design point, the design point  $\mathbf{U}^*$  is obtained from solving the following constrained optimization problem:

$$\text{Minimize} \quad O_7(\mathbf{U}) = \|\mathbf{U}\| = \left( \sum_{i=1}^{NRV} U_i^2 \right)^{1/2} \quad (48)$$

$$\text{Subject to constraint} \quad g(\mathbf{U}) = 0 \quad (49)$$

in which  $\mathbf{U} = [U_1 \dots U_{NRV}]^T$  denotes the vector of standard normal variables.  $g(\mathbf{U})$  is the performance function.  $g(\mathbf{U}) = 0$  denotes the limit state surface and  $g(\mathbf{U}) \leq 0$  indicates the failure state corresponding to the performance function. The equality constraint is modified to an inequality constraint

$$|g(\mathbf{U})| \leq \varepsilon \quad (50)$$

, or

$$G(\mathbf{U}) \leq 0 \quad (51.1)$$

, where

$$G(\mathbf{U}) = |g(\mathbf{U})| - \varepsilon \quad (51.2)$$

in which  $\varepsilon$  is the tolerance and set to a small value, e.g. 0.01.

4.4.2 Numerical example 4

Consider a parabolic performance function as introduced in (Kiureghian & Dakessian, 1998):

$$g(X_1, X_2) = b - X_2 - \kappa(X_1 - e)^2$$

(52)

where  $b$ ,  $\kappa$ , and  $e$  are deterministic parameters.  $X_1$  and  $X_2$  are standard normal variables. In this example,  $b = 5$ ,  $\kappa = 0.5$  and  $e = 0.1$ . This set of parameters leads to two design points.

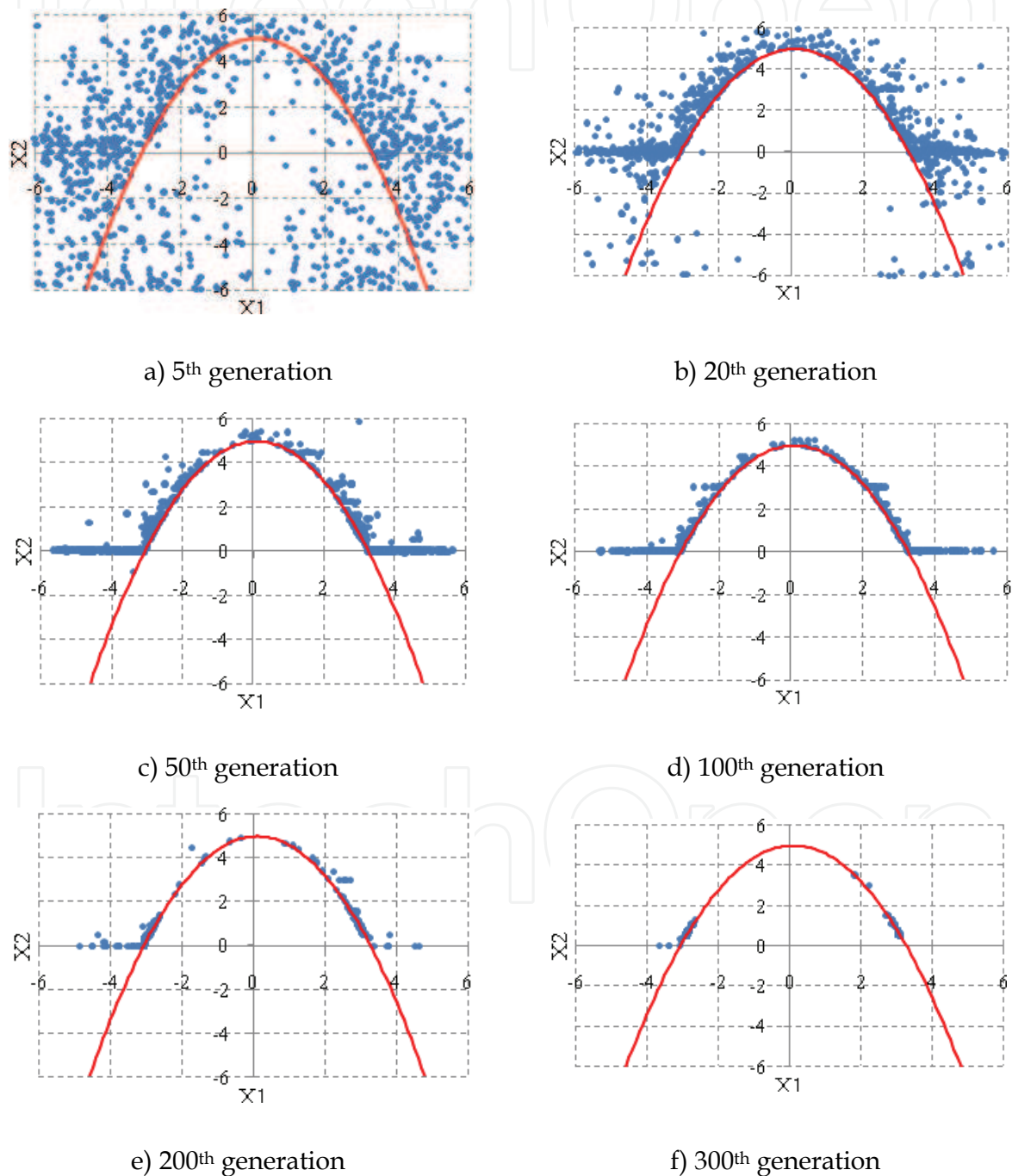


Fig. 4. Chromosomes distribution at various generations by DCGAS for two design points problem.

DCGAs are employed to determine both design points. The tolerance parameter  $\varepsilon$  is set to 0.01. The parameters of the DCGAs are given in Table 5. The fitness function is

$$F(\boldsymbol{U}) = \begin{cases} 1/O_8(\boldsymbol{U}) & ; G(\boldsymbol{U}) \leq 0 \\ 1/[O_8(\boldsymbol{U}) + k\nu(\boldsymbol{U})] & ; \text{otherwise} \end{cases} \tag{53}$$

, where

$$O_8(\boldsymbol{U}) = O_8(X_1, X_2) = [X_1^2 + X_2^2]^{1/2} \tag{54}$$

$$G(\boldsymbol{U}) = G(X_1, X_2) = |b - X_2 - \kappa(X_1 - e)^2| - \varepsilon \tag{55}$$

$$\nu(X_1, X_2) = \begin{cases} |G(X_1, X_2)| & ; G(X_1, X_2) > 0 \\ 0 & ; \text{otherwise} \end{cases} \tag{56}$$

Parameters	Value
Population Size	1000
Crossover Probability	1.0
Number of Executed Generations	300

Table 5. DCGAs parameters.

The distributions of chromosomes at consecutive generations by DCGAs are displayed in Figure 4. The solutions after the first generation are totally spread over the search area in the beginning. The solutions are then assembled in the parabolic topology during the course of optimization. After consecutive numbers of generations, the chromosomes gradually accumulate at two distinct design points. The objective function values, the optima or design points, and their corresponding distances to the origin are shown in Table 6. The numerical results are compared with the results from the literature (Kiureghian & Dakessian, 1998) in Table 6. The search method used in (Kiureghian & Dakessian, 1998) belongs to the class of gradient-based search and will be herein referred to as Gradient-based Search with Incrementally Added Constraint (GSIAC).

Optima	Method	$\boldsymbol{U}^* = (X_1^*, X_2^*)$	$g(X_1^*, X_2^*)$	$O_7(X_1^*, X_2^*)$
1	DCGAS	(-2.77, 0.88)	1.55x10 <sup>-3</sup>	2.91
	GSIAC	(-2.74, 0.97)	-2.80x10 <sup>-3</sup>	2.91
2	DCGAS	(2.83, 1.27)	3.55x10 <sup>-3</sup>	3.10
	GSIAC	(2.92, 1.04)	-1.62x10 <sup>-2</sup>	3.10

Table 6. Comparison of design points and their respective safety indices of two design points from DCGAS and the literature (Kiureghian & Dakessian, 1998).

It is clear that DCGAs yield the design points that are close to the results from GSIAC. It should be noted that the multimodal GAs work in a different manner from such a sequential search method as GSIAC, where the decision on the numbers of the desired design points must be made by the user. The population-based operation of GAs makes the search circumvent the problem of selecting appropriate starting search point, as appeared in the

gradient-based methods. In addition, the fundamental mechanisms of multimodal GAs are able to automatically detect and capture several design points.

The whole operation of multimodal GAs shows that a priori knowledge about the geometry of performance function is not required. This makes GAs operable to practical problems where the geometry of performance functions cannot be generally visualized. Therefore, the capability in PRA of multiple failure events with almost equal levels of likelihood can be enhanced by using GAs.

## 5. Conclusion

The enhancement of capability in Probabilistic Risk Analysis (PRA) by Genetic Algorithms (GAs) is described. Several key aspects of PRA that are enhanced by GAs include the determination of Point of Maximum Likelihood (PML) in failure domain, Monte Carlo Simulation (MCS)-based PRA of rare events under highly constrained computational resources, the improvement of confidence levels in PRA results by applying GAS-determined suboptimal Importance Sampling Functions (ISFs), and the automatic and simultaneous detection of multiple failure modes with almost equal likelihoods. All of these achievements are attributed to the problem knowledge-free operation of GAs. This feature of capability enhancement is testified via numerical examples where complicate and thus non-visualizable performance functions as well as mixtures of different random variables are considered. Consequently, the capability in PRA is naturally enhanced to practical problems.

The present application of GAs to PRA is limited to the uncertainty of aleatory type. Future application of GAs will be extended to the uncertainty of epistemic type or the combination of both types. Such extended application of GAs will enhance the capability in PRA to the cases where expert opinions and their corresponding degrees of belief are included in the analysis. The analysis then becomes more rationale and realistic. Since DCGAs exhibit genetic drift, it is also beneficial to develop novel multimodal GAs that reduce the genetic drift and increase the search stability in the future. The algorithms to be developed should eliminate or require least a priori knowledge about the search space so that the algorithms are efficiently and effectively applicable to practical problems.

## 6. References

- Barbosa, H. & Lemonge, A. (2003). A new adaptive penalty scheme for genetic algorithms. *Information sciences*, 156, 215-251.
- Brownlee, J. (2004). *Parallel Niching Genetic Algorithms: A Crowding Perspective*, Thesis in Master of Information Technology, Centre for Intelligent Systems and Complex Processes, School of Information Technology, Swinburne University of Technology, Australia.
- Deb, K. (1995). *Optimization for Engineering Design Algorithms and Examples*, Prentice-Hall, New York.
- De Jong, K.A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*, Ph.D. Thesis, Department of Computer Science, University of Michigan, Ann Arbor, MI, USA.
- Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York.

- Gen, M. & Cheng, R. (1997). *Genetic algorithms and engineering design*. John Wiley and Sons, ISBN : 978-0-471-12741-3, New York
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Massachusetts.
- Harnpornchai, N. (2007). Determination of Suboptimal Importance Sampling Functions by Genetic Algorithms, *Proceedings of the European Safety and Reliability Conference 2007, ESREL 2007 - Risk, Reliability and Societal Safety 2*, pp. 1169-1176, Stavanger, Norway, 25-27 June 2007.
- Harnpornchai, N.; Chakpitak, N.; Chandarasupsang, T.; Tuang-Ath Chaikijkosi. & Dahal, K. (2007). Dynamic adjustment of age distribution in Human Resource Management by genetic algorithms, *Evolutionary Computation, CEC 2007, IEEE Congress on 25-28 Sept. 2007*, 1234 - 1239.
- Kiureghian, A.D. & Dakessian, T. (1998). Multiple Design Points in First and Second-Order Reliability, *Structural Safety*. 20, 37-49.
- Lee, S.M.; Chang, Y.S.; Choi, J.B. & Kim, Y.J. (2006). Failure probability assessment of wall-thinned nuclear pipes using probabilistic fracture mechanics, *Nuclear Engineering and Design*, 236, 350-358.
- Mahfoud, S.W. (1995a). *Niching methods for genetic algorithms*, Ph.D. dissertation, Univ. of Illinois, Urbana-Champaign.
- Mahfoud, S.W. (1995b). A comparison of parallel and sequential niching methods, *International Conference on Genetic Algorithms*, 136-143.
- Michalewicz, A. (1996). *Genetic + Data Structures = Evolution Programs*, second ed., Springer, ISBN : 978-3-540-60676-5, second ed. Berlin
- Miller, B.L & Shaw, M.J. (1995). Genetic algorithms with dynamic niche sharing for multi-modal function optimization, *IlliGAL Report no. 9510*.
- Obadage, A.S. & Harnpornchai, N. (2006). Determination of point of maximum likelihood in failure domain using genetic algorithms, *Int J Press Vessels Pipe*, 83, 4, 276-82.
- Tada, H. (1978). *The stress analysis of cracks handbook*, Del Research Corporation.

IntechOpen



## **Evolutionary Computation**

Edited by Wellington Pinheiro dos Santos

ISBN 978-953-307-008-7

Hard cover, 572 pages

**Publisher** InTech

**Published online** 01, October, 2009

**Published in print edition** October, 2009

This book presents several recent advances on Evolutionary Computation, specially evolution-based optimization methods and hybrid algorithms for several applications, from optimization and learning to pattern recognition and bioinformatics. This book also presents new algorithms based on several analogies and metafores, where one of them is based on philosophy, specifically on the philosophy of praxis and dialectics. In this book it is also presented interesting applications on bioinformatics, specially the use of particle swarms to discover gene expression patterns in DNA microarrays. Therefore, this book features representative work on the field of evolutionary computation and applied sciences. The intended audience is graduate, undergraduate, researchers, and anyone who wishes to become familiar with the latest research work on this field.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Napat Harnpornchai (2009). Enhancement of Capability in Probabilistic Risk Analysis by Genetic Algorithms, Evolutionary Computation, Wellington Pinheiro dos Santos (Ed.), ISBN: 978-953-307-008-7, InTech, Available from: <http://www.intechopen.com/books/evolutionary-computation/enhancement-of-capability-in-probabilistic-risk-analysis-by-genetic-algorithms>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen